

REMARKS

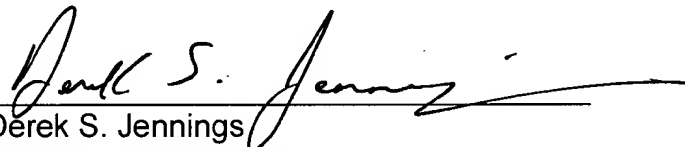
Amended claims 2 – 8, 10 – 13 and 15 have been re-written to place them in better form for examination. No new matter has been added.

Attached hereto is a marked-up copy of the amended claims 2 – 8, 10 – 13 and 15.

Further attached hereto are clean copies of the amended 2 – 8, 10 – 13 and 15.

Please charge any fee necessary to enter this paper and any previous paper to deposit account 09-0468.

Respectfully submitted,

By: 
Derek S. Jennings
Registered Patent Agent / Patent Engineer
Reg. No. 41,473

IBM Corporation
Intellectual Property Law Department
P. O. Box 218
Yorktown Heights, New York 10598
Telephone No.: (914) 945-2144

MARKED-UP COPY OF THE AMENDED CLAIMS

2. The method of claim 1, ~~wherein~~ further comprising the step of providing said predefined area of storage is ~~in~~with volatile memory.
3. The method of claim 1, ~~wherein~~ further comprising the step of providing said predefined area of storage is ~~in~~with non-volatile memory.
4. The method of claim 1, further comprising the step of:
outputting the contents of said storage area to a file at a predetermined point in time.
5. The method of claim 4, further comprising the step of:
outputting trace information to said file upon exit from said at least one module.
6. The method of claim 5, ~~wherein~~ further comprising the step of outputting the contents of said storage area is ~~outputted~~ at the same time as said exit trace information.
7. The method of claim 4, wherein the step of outputting the contents of said storage area ~~comprises:~~ further comprises the step of:
determining whether said storage area is full; and
responsive to a positive determination, outputting said contents to said file.
8. The method of claim 4, wherein the step of outputting the contents of said storage area ~~comprises:~~ further comprises the step of:
determining whether a failure has occurred within said program; and
responsive to a positive determination, outputting said contents to said file.
10. The method of claim 9, further comprising the step of:

writing the position of the most recent unique identifier to be written out to said storage area to said storage area.

11. The method of claim 10, ~~wherein~~ further comprising the step of using said position ~~is used~~ to determine the number of unique identifiers that have been overwritten prior to being written out to said file.

12. The method of claim 11, further comprising the step of:
responsive to determining that a large number of unique identifiers have been overwritten, increasing the size of said predefined area of storage.

13. An apparatus for tracing the execution path of a computer program comprising at least one module including a plurality of instructions, at least one of said instructions being a branch instruction, said apparatus comprising:

~~means~~ an identifier for identifying each branch instruction;

~~means~~ an evaluator for evaluating each branch instruction to be one of true and false; and

~~means~~ a pusher, responsive to an evaluation of true, for pushing a unique identifier into a predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true.

15. A compiler for instrumenting a computer program comprising at least one module including a plurality of instructions, at least one of said instructions being a branch instruction, each branch instruction being evaluated to be one of true and false at run-time, with at least one signature instruction for indicating the execution path of said program at run-time, said compiler comprising:

~~means~~ a first identifier for identifying each branch instruction;

~~means~~ a second identifier for identifying the instructions associated with an evaluation of true at run-time;

~~means~~ a pusher for instrumenting said instructions associated with an evaluation of true with a signature instruction, wherein said signature instruction causes a unique identifier to be pushed into a predefined area of storage upon execution of said true instructions at run-time.